

# 实验二 移位寄存器实验

## 实验目的

- 1. 掌握移位寄存器的工作原理
- 2. 掌握 Modelsim 软件的使用方法
- 3. 掌握 Vivado 软件的开发流程；锻炼使用 Verilog 语言描述硬件的能力
- 4. 熟悉移位寄存器的 HDL 描述方法，理解其工作特性；

## 实验原理

实验原理图如图 1-1，长方形框内是在 FPGA 内部设计的实验电路。定义了 2 个双向移位寄存器 194、输入数据 DATA，由时钟 CLK、Reset、S1、S0 分别移位寄存器的操作。

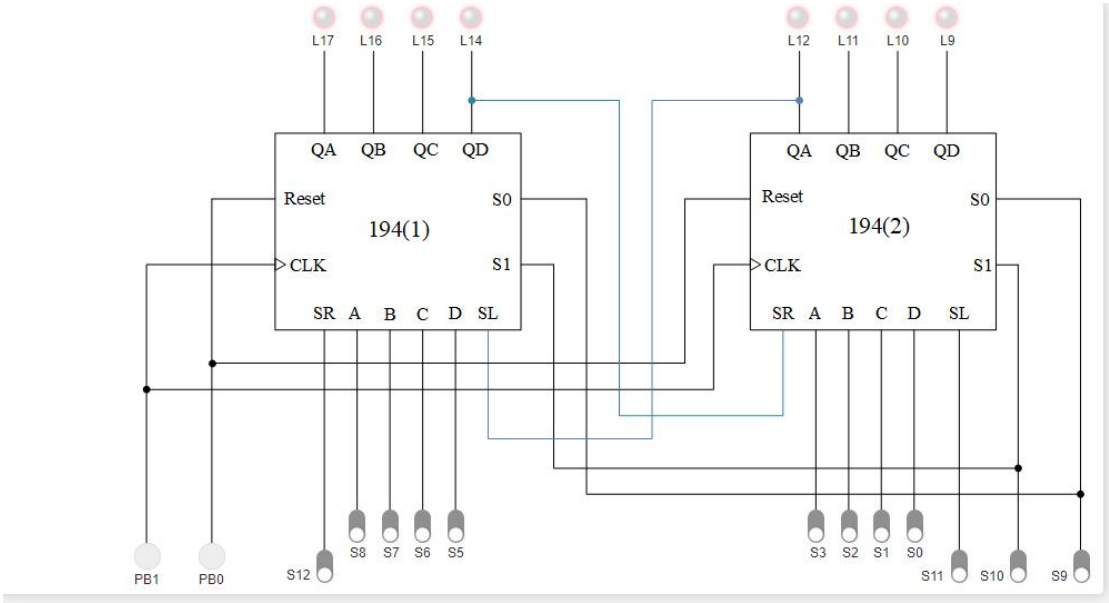


图 1-1 实验二虚拟面板示意图








## 实验内容

1. **基础实验。**验证功能，并操作分析、记录结果。

利用 2 片 194 实现一个 8 位双向移位寄存器，并在虚拟面板上验证移位寄存器的功能(接收数据、左移、右移)，并操作分析、记录结果。

(1) 移位寄存器的特性

|    | DATA  |       | SR | SL | S1 | S0 | CLK | RESET | SR_Q    |        |
|----|-------|-------|----|----|----|----|-----|-------|---------|--------|
|    | S8-S5 | S4-S0 |    |    |    |    |     |       | L17-L14 | L12-L9 |
| 复位 |       |       | —  |    | —  | —  | —   |       |         |        |
| 直  |       |       | 0  | 0  | 0  | 0  |     | 0     |         |        |

|    |  |   |   |   |   |   |   |  |
|----|--|---|---|---|---|---|---|--|
| 送  |  | 1 | 1 | 1 | 1 |  | 0 |  |
| 左移 |  | 0 | 0 | 1 | 0 |  | 0 |  |
|    |  | 1 | 1 | 1 | 0 |  | 0 |  |
| 右移 |  | 0 | 0 | 0 | 1 |  | 0 |  |
|    |  | 1 | 1 | 0 | 1 |  | 0 |  |
| 锁存 |  | 0 | 0 | 0 | 0 |  | 0 |  |
|    |  | 1 | 1 | 1 | 1 |  | 0 |  |

实验现象分析：

- ① 直送是将输入端数据装入到移位寄存器，既不左移也不右移。
- ② 左移和右移的信号分别是\_\_\_\_\_和\_\_\_\_\_。
- ③ 本实验设计的移位寄存器是对移位寄存器进行\_\_\_\_\_移位。

2. 提高实验

（1）用 Verilog 硬件描述语言实现一个 2 选 1 多路选择器 MUX2\_1、4 选 1 多路选择器 MUX4\_74153，并用 Modelsim 软件进行仿真测试。

- ① 说明：  
通过 BA 控制输出 Y 选择 D0-D3
- ② 代码：

```

`timescale 1ns / 1ps

module mux4_74153 (G, B, A,
                  D0, D1, D2, D3, Y);
    input G, B, A;
    input D0, D1, D2, D3;
    output Y;

endmodule

③ 仿真波形：

```

（2）设计一个 4 位双向移位寄存器（SR194），并用 Modelsim 软件进行仿真测试。另外在 Vivado 环境下进行综合、仿真、调试，并下载到 Digilent Nexys4 FPGA 开发板进行验证。

4 位双向移位寄存器（SR194），功能类似芯片 74LS194，具体如下：

|     |      |       |           |
|-----|------|-------|-----------|
| 类 别 | 电路简称 | 真 值 表 | 电 路 引 脚 图 |
|-----|------|-------|-----------|

| 4 位 并<br>入 并 出<br>移 位 寄<br>存 器 | 7419<br>4 | 输 入 |       |     |    |    | 输 出       |
|--------------------------------|-----------|-----|-------|-----|----|----|-----------|
|                                |           | CLR | S1 S0 | CLK | SL | SR | QA~QD     |
|                                |           | 0   | × ×   | ×   | ×  | ×  | 清除，全 0    |
|                                |           | 1   | 0 0   | 0   | ×  | ×  | 禁止时钟，保持   |
|                                |           | 1   | 1 1   | ↑   | ×  | ×  | 并接 A~D，置数 |
|                                |           | 1   | 0 1   | ↑   | ×  | 0  | 右移，移入 0   |
|                                |           | 1   | 0 1   | ↑   | ×  | 1  | 右移，移入 1   |
|                                |           | 1   | 1 0   | ↑   | 0  | ×  | 右移：QA→QD  |
|                                |           | 1   | 1 0   | ↑   | 1  | ×  | 左移，移入 0   |
|                                |           |     |       |     |    |    | 左移，移入 1   |
|                                |           |     |       |     |    |    | 左移：QA←QD  |

图 1-2 四位移位寄存器引脚图

说明：

① CLR 为清零信号，为 0 时 QA-QD 全置 0

CLK 时钟上升沿时通过 S1、S0、SL、SR 控制输出信号 QA-QD。S1S0 为 11 时把输入 A-D 写入 QA-QD；S1S0 为 01 时右移 QA→QD，SR 为 0 时 QA 移入 0，SR 为 1 时 QA 移入 1；S1S0 为 10 时左移 QA←QD，SL 为 0 时 QD 移入 0，SL 为 1 时 QD 移入 1。

② 部分代码：

```
`timescale 1ns / 1ps
```

```
module REG_74194 (CLR, S1, S0, CLK, SL, SR, A, B, C, D, QA, QB, QC, QD);
```

```
    input CLR, S1, S0, CLK, SL, SR;
```

```
    input A, B, C, D;
```

```
    output QA, QB, QC, QD;
```

```
    reg QA, QB, QC, QD;
```

```
endmodule
```

③ 仿真波形：

(3) 利用 (2) 设计的 2 个双向 4 位移位寄存器 SR194（功能类似 74LS194），实现一个双向 8 位移位寄存器，用 Modelsim 软件进行仿真测试，并利用 Vivado 平台进行综合、仿真、调试，下载到 Digilent Nexys4 FPGA 开发板进行验证，接收开关的数据输入，并用指示灯观察数据的输出。

具体连线方法是：将其中一片 SR194 的 Q<sub>D</sub> 接到 SR 端，而将另一片的 Q<sub>A</sub> 接到 SL 端，同时把两片的 S<sub>1</sub>、S<sub>0</sub>、CLK、CLR 端分别并联起来。如下图所示。

说明：A、B、C、D、SR、SL、S1、S0 接开关，CLK、CLR 接按键，Q<sub>A</sub>、Q<sub>B</sub>、Q<sub>C</sub>、Q<sub>D</sub> 接指示灯。

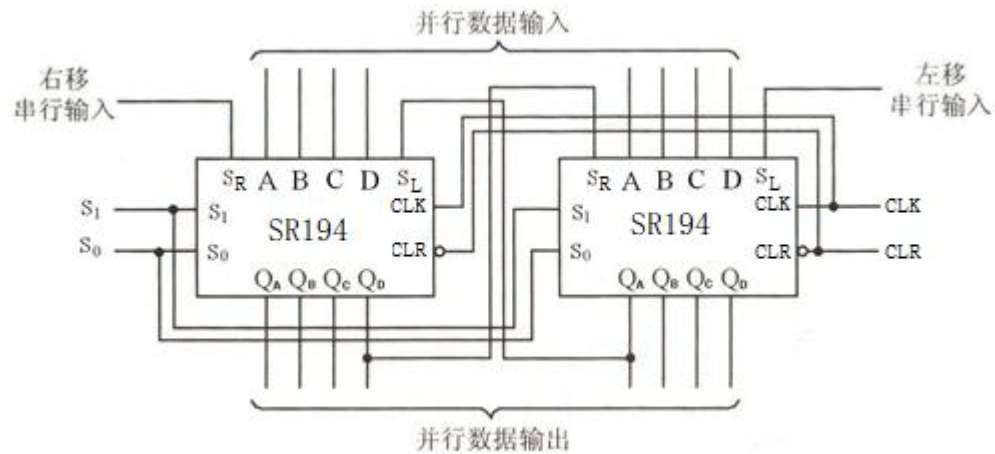


图 1-3 8 位移位寄存器的电路图

虚拟面板上输入输出分配如下：

输入：CLK: PB1

RESET (CLR) :PB0

右边寄存器输入 S[3]~S[0] (A~D)

左边寄存器输入 S[8]~S[5] (A~D)

S1,S0: S[10],S[9]

SR ,SL:S[12],S[11]

输出：左边寄存器输出：LED[17:14]

右边寄存器输出：LED[12:9]